
Abstract
ET: Enhanced tool
The effectiveness of <a graphical grid of data cells linked by textual formulas> in supporting the design of <spreadsheets> has been demonstrated. An enhanced method is described for the design of <spreadsheets> based on <allowing end users to define reusable functions without leaving the grid-and-text paradigm>. Examples are provided confirming the effectiveness of its support for <spreadsheets> in design.

Why ET?
Spreadsheets had been around for over a decade. This paper constitutes an improved method of supporting the creation of spreadsheet programs by end users.

Question - [Method/means of development]
<model-type / solution strategy>
What is the best way to bring the benefits of traditional programming languages with reusability constructs to spreadsheet development—specifically, improved maintainability, improved encapsulation of expertise (and therefore constructability), and improved performance?

<artifact-type>
The artifact being produced is a spreadsheet. The question is the best way to help end users create spreadsheets with the attributes listed above.

Results - [Tool / Notation]
<model-type / solution strategy>
The authors argue that functions constitute a powerful form of reuse in traditional languages, and that offering functions to spreadsheet developers will bring the concomitant benefits listed above.

Validation – [Evaluation against Empirical Model]
<model-type / solution strategy>
The authors correctly note that one significant question is whether end users will adopt function support. After all, Excel supported macros since about a decade prior to this paper, and by all accounts, virtually nobody uses macros extensively. The main goal of macros was to allow spreadsheet creators to create reusable functions. So the authors take great pains to ensure that their approach does not fall prey to the same low adoption.

They attempt to achieve this by explicitly using Green’s classic Cognitive Dimensions to guide and evaluate their design. Perhaps the most significant message that they take from these Dimensions is it is absolutely essential to retain the traditional grid-and-textual-formula paradigm inherent to all extant spreadsheet design environments, since that is the only programming paradigm that most end users know.

This choice has certain somewhat distressing side-effects, including the need to visualize activation records as spreadsheets, which will probably feel unintuitive to traditional programmers. Moreover, as of the paper’s date, they had not actually implemented the bulk of their proposed functionality in Excel, but rather an experimental system called Forms/3. (They did implement a small portion in Excel and did PowerPoint mock-ups of other functions.) They have utilized these prototypes for small examples, but they have not undertaken formal empirical validation in the usual HCI sense.